



Rapport de stage de fin d'études de Master

Stage en informatique – Modélisation des données et
IA

Théo JOLY

Master DataScale – Université Paris-Saclay / UVSQ

Réalisé au sein d'Ecolab, laboratoire d'innovation du CGDD

Tuteur académique : Yehia Taher

Tuteurs de stage : Clément Jaquemet et Alexis Leautier

Année universitaire 2025–2026

Remerciements

Je remercie mon tuteur académique, **Yehia Taher**, qui assure le suivi administratif et l'évaluation de ce stage de fin d'études.

J'exprime ma profonde gratitude envers mes tuteurs de stage, **Clément Jaquemet** et **Alexis Leautier**, pour leur encadrement de qualité, leur disponibilité et les nombreux échanges enrichissants qui m'ont permis de progresser tant sur le plan technique que professionnel.

Je tiens également à remercier chaleureusement **Hélène Begon** pour son aide précieuse dans les démarches administratives liées à mon arrivée, notamment pour la signature de ma convention de stage, la demande de matériel informatique, l'obtention de mon badge d'accès et l'ensemble de la logistique mise en place.

Je remercie l'ensemble de l'équipe Ecolab pour son accueil chaleureux, sa bienveillance et l'atmosphère de travail agréable qu'elle a su instaurer.

Résumé

Ce stage de fin d'études de Master DataScale a été réalisé au sein d'Ecolab, le laboratoire de l'innovation au service de la transition écologique, rattaché au Commissariat général au développement durable (CGDD). L'objectif était d'évaluer l'usage de l'intelligence artificielle pour la standardisation et la mise en cohérence de modélisations de données, dans le contexte du Conseil national de l'information géolocalisée (CNIG). À partir du prototype **AI4Semantics** de la Commission européenne, développé dans le cadre de SEMIC (*Semantic Interoperability Community*), j'ai développé une preuve de concept nommée HarmonIA, fondée sur une architecture RAG hybride (embeddings denses et sparse avec reranking), une base vectorielle Qdrant et un modèle de langage hébergé de manière souveraine via Ollama Cloud. L'outil intègre les standards du CNIG, les schémas de `schema.data.gouv.fr` et plusieurs référentiels sémantiques internationaux. Il propose désormais une interface de modélisation interactive via le protocole MCP, au-delà d'un simple chatbot. Les tests menés auprès de partenaires institutionnels, notamment dans le cadre du projet **Building Data X** (espace de données construction et immobilier porté par la Caisse des Dépôts, auquel Ecolab contribue), ont confirmé l'intérêt de la démarche et ouvert des perspectives d'industrialisation.

Mots-clés : intelligence artificielle, RAG, interopérabilité sémantique, standards de données, CNIG, modélisation de données, IA frugale.

Table des matières

Remerciements	2
Résumé	3
Introduction générale	6
0.1 Contexte et motivation	6
0.2 Objectifs du stage	6
0.3 Organisation du rapport	7
1 Contexte du stage	8
1.1 Présentation de l'entité d'accueil : Ecolab	8
1.2 Projets en cours et environnement technique	8
2 Objectif du stage	9
2.1 Présentation du projet d'accueil	9
2.2 Mission confiée au stagiaire	9
2.3 Spécifications et contraintes	10
3 État de l'art	11
3.1 Interopérabilité sémantique et standardisation des données	11
3.2 Grands modèles de langage (LLM)	11
3.3 Génération augmentée par récupération (RAG)	12
3.4 Bases de données vectorielles et embeddings	12
3.5 Reranking	13
3.6 Agents et outils : le protocole MCP	13
3.7 IA frugale et souveraineté des données	13
3.8 Hallucination des modèles de langage	14
3.9 Ontology matching et grands modèles de langage	14
3.10 Génération augmentée et validation sémantique	14
3.11 Outils existants	14
3.12 Positionnement du stage	15
4 Approche suivie et solution proposée	16
4.1 Analyse du besoin	16
4.2 Conception de la solution	16
4.3 Développement et implémentation	19

5	Validation	25
5.1	Protocole et résultats de validation	25
5.2	Analyse et discussion	27
6	Conclusion et perspectives	29
6.1	Bilan du stage	29
6.2	Perspectives	29
	Bibliographie	32
	Annexes	33
.1	Annexe A : Codes source / scripts complémentaires	33
.2	Annexe B : Liste des figures techniques	33
.3	Annexe C : Résultats détaillés	33

Introduction générale

0.1 Contexte et motivation

Dans un contexte où les données territoriales et environnementales occupent une place croissante dans la conception des politiques publiques, la capacité à produire, partager et réutiliser des modèles de données interopérables constitue un enjeu majeur. Le Conseil national de l'information géolocalisée (CNIG) anime en France un travail de standardisation visant à harmoniser la modélisation des données géographiques. Cependant, ces standards sont majoritairement publiés sous forme de documents PDF, ce qui limite leur exploitation automatique et leur alignement avec d'autres référentiels existants.

Par ailleurs, l'émergence des grands modèles de langage (LLM) et des architectures de génération augmentée par récupération (RAG) offre de nouvelles perspectives pour assister les groupes de travail en charge de la standardisation. L'objectif est de permettre une meilleure circulation entre les modèles de données, de faciliter leur comparaison et de réduire les redondances, tout en maîtrisant l'impact environnemental des technologies utilisées. Ce rapport présente le stage réalisé au sein d'Ecolab, le laboratoire de l'innovation du Commissariat général au développement durable, et la preuve de concept d'assistant IA nommée HarmonIA, conçue pour accompagner les groupes de travail du CNIG dans leurs tâches de modélisation.

0.2 Objectifs du stage

Le stage intitulé *Stage en informatique – Modélisation des données et IA* avait pour objectif d'évaluer l'utilisation de l'intelligence artificielle pour la standardisation et la mise en cohérence de modélisations de données variées. L'ambition était de fournir les éléments de décision nécessaires au développement d'un outil d'IA opérationnel destiné aux groupes de travail du CNIG.

Ma mission personnelle consistait à :

- réaliser une étude de la littérature scientifique et un comparatif des outils IA disponibles ;
- m'approprier le code de l'outil AI4Semantics de la Commission européenne ;
- traiter et intégrer les standards de données du CNIG via une architecture RAG ;
- expérimenter des LLM pour augmenter les connexions sémantiques entre standards existants ;
- conduire des discussions et ateliers avec des partenaires institutionnels si les expé-

rimentations le nécessitaient.

0.3 Organisation du rapport

Ce rapport est structuré en six chapitres. Le premier présente le contexte du stage et l'entité d'accueil. Le deuxième détaille les objectifs et la mission confiée. Le troisième propose un état de l'art sur les techniques mobilisées. Le quatrième décrit l'approche suivie et la solution développée. Le cinquième présente la validation et les résultats obtenus. Enfin, le sixième chapitre dresse le bilan du stage et ouvre des perspectives.

Chapitre 1

Contexte du stage

1.1 Présentation de l’entité d’accueil : Ecolab

J’ai effectué mon stage au sein d’**Ecolab**, le laboratoire d’innovation au service de la transition écologique du Commissariat général au développement durable (CGDD). Ecolab anime l’écosystème **Greentech Innovation**, assure le rôle d’administrateur ministériel des données, des algorithmes et des codes sources (AMDAC) pour le CGDD, et héberge le secrétariat du **Conseil national de l’information géolocalisée** (CNIG). Il est situé à La Défense, dans la tour Séquoia. J’ai été intégré à l’équipe **Data/IA**, qui explore les usages de la donnée et de l’intelligence artificielle pour accompagner les politiques publiques environnementales.

1.2 Projets en cours et environnement technique

Le stage s’est inscrit dans un projet porté par Ecolab autour de la sémantique et de l’interopérabilité des modèles de données. L’objectif était d’explorer comment l’IA pouvait faciliter le travail de standardisation du CNIG.

L’environnement technique du stage comprenait :

- des **modèles de langage (LLM)**, notamment **gemma4 31b** hébergé via Ollama Cloud ;
- une architecture **RAG** (Retrieval-Augmented Generation) pour enrichir les réponses du modèle avec des documents standards ;
- des **modèles d’embeddings** denses et sparse, en l’occurrence **bge-m3**, couplés à un modèle de *reranking* **bge-reranker-v2-m3** ;
- une base de données vectorielle **Qdrant** ;
- le déploiement sur **SSPCloud Onyxia**, la plateforme cloud des services statistiques publics ;
- une base de code initiale issue de l’outil **AI4Semantics** de la Commission européenne, réutilisée et adaptée dans le cadre du stage.

Chapitre 2

Objectif du stage

2.1 Présentation du projet d'accueil

Le projet sur lequel j'ai travaillé visait à concevoir une preuve de concept d'outil d'IA pour accompagner la standardisation et la mise en cohérence de modélisations de données. Le point de départ était **AI4Semantics**, un prototype de chatbot augmenté par un LLM, développé par la Commission européenne dans le cadre de **SEMIC** (*Semantic Interoperability Community*). SEMIC rassemble une communauté d'adoptants autour de standards sémantiques comme les *Core Vocabularies*, DCAT-AP et les validateurs sémantiques.

Le besoin exprimé par Ecolab était d'adapter AI4Semantics au contexte français, en intégrant les standards du CNIG, les schémas de données de `schema.data.gouv.fr` et d'autres référentiels. L'objectif était de fournir aux groupes de travail du CNIG un outil capable de réutiliser les modélisations existantes, de limiter la redondance et d'améliorer la cohérence des nouveaux standards. Le CNIG produit des préconisations nationales dans des domaines variés (urbanisme, adresse, accessibilité, occupation du sol, réseaux, etc.), qui deviennent obligatoires lorsque la réglementation y fait référence. Leur élaboration suit la **Fabrique des standards**, un processus collaboratif long et mobilisant de nombreux experts. C'est dans ce contexte que l'IA peut apporter une aide significative, en accélérant l'analyse de l'existant et en facilitant l'alignement avec d'autres référentiels.

2.2 Mission confiée au stagiaire

Ma mission principale consistait à évaluer la faisabilité et l'intérêt d'un tel outil, puis à en produire une preuve de concept fonctionnelle. Plus précisément, mes responsabilités comprenaient :

- la réalisation d'une revue de littérature et d'un comparatif des outils IA existants pour la modélisation de données ;
- l'appropriation du code de l'outil AI4Semantics et son adaptation aux besoins du projet ;
- le traitement, le découpage et l'intégration des standards du CNIG dans une architecture RAG ;
- l'enrichissement du corpus avec d'autres référentiels : `schema.data.gouv.fr`, LOV,

INSPIRE, FIWARE, schema.org, SAREF et SEMIC ;

- l’expérimentation de LLM pour augmenter les connexions sémantiques entre standards et faciliter la création de modèles de données ;
- la conduite d’échanges avec des partenaires institutionnels, notamment la DINUM, OpenData France, des acteurs du projet **Building Data X** (Caisse des Dépôts, Orange Business, Dawex, Docaposte, Sopra Steria, buildingSmart France, FPI), l’AFNOR et Inria.

J’étais encadré au quotidien par **Clément Jaquemet**, tuteur officiel, et par **Alexis Leautier**, second tuteur. Un point hebdomadaire était organisé tous les mardis matin pour faire le bilan de l’avancée et définir les pistes à explorer. Par ailleurs, l’équipe Ecolab tenait une réunion hebdomadaire tous les jeudis matin, durant laquelle les membres présentaient leurs travaux, des projets en cours ou des notions techniques partagées.

2.3 Spécifications et contraintes

Le stage s’est déroulé en mode hybride : quatre jours de présentiel à La Défense et un jour de télétravail par semaine. Cette organisation a permis une bonne intégration dans l’équipe tout en conservant une certaine flexibilité.

Les principales contraintes du projet étaient les suivantes :

- **Contrainte documentaire** : les standards du CNIG sont majoritairement disponibles sous forme de fichiers PDF, ce qui rend leur exploitation automatique difficile. L’outil devait pouvoir ingérer et traiter ces documents de manière robuste.
- **Contrainte d’interopérabilité** : l’outil devait s’appuyer sur des référentiels variés et hétérogènes, en assurant leur cohabitation dans un même système de recherche sémantique.
- **Contrainte de souveraineté et de confidentialité** : les données traitées étant relatives à des référentiels publics et pouvant inclure des éléments sensibles, le modèle de langage choisi devait être hébergé en Europe et ne pas conserver les requêtes. C’est pourquoi **gemma4 31b** a été déployé via Ollama Cloud.
- **Contrainte de frugalité** : dans la lignée des engagements d’Ecolab, les choix techniques devaient privilégier des solutions sobres, notamment via l’utilisation de modèles compacts et l’optimisation des échanges avec le modèle.
- **Contrainte temporelle** : le stage, commencé en avril 2026, devait aboutir à une preuve de concept testable et documentée avant la fin du stage en septembre 2026.

Chapitre 3

État de l’art

3.1 Interopérabilité sémantique et standardisation des données

L’interopérabilité sémantique désigne la capacité de systèmes d’information différents à échanger des données de manière non seulement syntaxique mais aussi sémantique, c’est-à-dire en partageant la signification des concepts manipulés. Dans le domaine de l’information géographique, cette interopérabilité repose sur la définition de standards de données communs, généralement exprimés sous la forme de modèles conceptuels, de schémas ou d’ontologies.

Ces standards sont élaborés au sein de groupes de travail thématiques et publiés sous forme de documents de référence. Le CNIG a formalisé cette démarche à travers la **Fabrique des standards**, un processus collaboratif dédié à l’élaboration, la maintenance et l’évolution des standards de données géolocalisées. Cependant, la multiplication des standards, leur format de publication (souvent PDF) et leur hétérogénéité rendent leur consultation, leur comparaison et leur alignement fastidieux. Cette situation appelle des outils capables de structurer, d’indexer et d’exploiter automatiquement ces référentiels.

3.2 Grands modèles de langage (LLM)

Les grands modèles de langage (LLM) sont des modèles de réseaux de neurones entraînés sur d’importants corpus textuels pour générer du langage naturel. Ils sont capables de résumer des textes, répondre à des questions, produire du code ou structurer des informations. Dans le domaine de la modélisation de données, ils peuvent aider à proposer des classes, des attributs, des relations et des définitions à partir d’un besoin exprimé en langage naturel.

Néanmoins, les LLM présentent plusieurs limites :

- **Hallucinations** : le modèle peut générer des informations plausibles mais incorrectes ou non fondées ;
- **Connaissance figée** : les connaissances du modèle sont limitées à ses données d’entraînement et ne reflètent pas les dernières évolutions des standards ;

- **Contexte limité** : chaque échange consomme des tokens, ce qui limite la quantité d'information injectée dans une conversation ;
- **Coût environnemental** : les modèles les plus massifs nécessitent des ressources computationnelles importantes, ce qui entre en tension avec les objectifs de sobriété numérique.

Ces limites justifient le recours à des architectures de génération augmentée par récupération (RAG) et à des modèles plus compacts.

3.3 Génération augmentée par récupération (RAG)

Le RAG consiste à enrichir la requête soumise au LLM avec des documents pertinents extraits d'une base documentaire. Le processus se déroule généralement en trois étapes : la récupération d'informations pertinentes, leur injection dans le contexte du modèle, puis la génération de la réponse. Cette approche permet de réduire les hallucinations en ancrant les réponses du modèle dans des sources documentaires, et de mettre à jour les connaissances du système sans réentraîner le modèle.

Dans le cadre de ce stage, le RAG permet d'indexer les standards du CNIG et d'autres référentiels pour qu'ils soient mobilisés lors de conversations de modélisation. Le modèle peut alors citer ses sources et s'appuyer sur des définitions et des structures existantes.

3.4 Bases de données vectorielles et embeddings

Pour mettre en œuvre un système RAG, les documents sont d'abord découpés en fragments (*chunks*), puis transformés en vecteurs numériques appelés **embeddings**. Ces vecteurs captent la signification sémantique du texte. Les requêtes des utilisateurs sont elles aussi encodées sous forme de vecteurs, et la base de données vectorielle recherche les fragments les plus proches dans l'espace vectoriel. La proximité est généralement mesurée par la similarité cosinus.

On distingue principalement deux types d'embeddings :

- **Embeddings denses** : représentations vectorielles compactes dans lesquelles chaque dimension porte une information sémantique latente. Ils sont performants pour capturer le sens global d'un texte.
- **Embeddings sparse** : représentations de grande dimension, souvent construites à partir de la fréquence des termes. Ils sont efficaces pour la correspondance lexicale exacte et la recherche par mot-clé.

L'utilisation conjointe de ces deux types d'embeddings permet de combiner la robustesse sémantique des embeddings denses et la précision lexicale des embeddings sparse.

3.5 Reranking

Le **reranking** est une étape de réordonnancement des documents récupérés par une première recherche vectorielle. Un modèle de reranking évalue la pertinence réelle de chaque document par rapport à la requête et remonte les résultats les plus adaptés. Cette étape améliore nettement la qualité des sources présentées au LLM, notamment lorsque le corpus est hétérogène et contient des documents aux formats variés.

Dans ce stage, le modèle **bge-reranker-v2-m3** a été utilisé pour affiner le classement des fragments de standards les plus pertinents avant leur injection dans le contexte du LLM.

3.6 Agents et outils : le protocole MCP

Les systèmes agentiques fondés sur les LLM peuvent être dotés d’**outils** leur permettant d’interagir avec des services externes, de manipuler des fichiers ou d’effectuer des opérations structurées. Le protocole **Model Context Protocol (MCP)**, popularisé par Anthropic, définit un standard pour connecter des assistants IA à des outils et des sources de données. Il permet de découpler le moteur de dialogue des capacités externes, facilitant ainsi l’intégration de nouvelles fonctionnalités.

Dans le cadre du stage, le protocole MCP a permis d’exposer plusieurs outils au modèle : un moteur de recherche sémantique, des fonctions d’édition de modèles (ajout de classes, d’attributs, de relations), et d’autres services complémentaires. Cette modularité a facilité le passage d’un simple chatbot à un outil de modélisation interactif.

3.7 IA frugale et souveraineté des données

L’IA frugale vise à réduire l’empreinte environnementale et économique des systèmes d’intelligence artificielle, notamment en privilégiant des modèles plus légers, des architectures efficaces et une optimisation des échanges. Cette démarche s’inscrit pleinement dans les engagements d’Ecolab, qui anime notamment la Sustainable AI Coalition.

La souveraineté des données est un autre enjeu important pour une administration publique. Le recours à des modèles hébergés en Europe et ne conservant pas les requêtes, comme c’est le cas avec Ollama Cloud pour le modèle **gemma4 31b**, permet de limiter les risques de fuite ou d’exploitation des données sensibles.

3.8 Hallucination des modèles de langage

L’hallucination est l’un des principaux obstacles au déploiement fiable des modèles de génération de langage naturel [5]. Elle se caractérise par la production d’un texte fluide mais factuellement incorrect. Dans le contexte de la modélisation de standards publics, une définition ou une relation erronée peut se propager dans des systèmes d’information. Les principales stratégies de mitigation passent par l’ancrage des réponses dans des documents de référence et l’utilisation de modèles capables de citer leurs sources.

3.9 Ontology matching et grands modèles de langage

L’**ontology matching** vise à identifier des correspondances entre entités de modélisations hétérogènes [4]. Hertling et Paulheim montrent qu’avec un prompt bien conçu et quelques exemples, un modèle ouvert peut égaler des systèmes supervisés [4]. Qiang et al. proposent Agent-OM, un framework agentique qui combine planification, recherche et mémoire pour l’alignement d’ontologies [10]. Zamazal explore le *complex ontology matching*, où une entité peut correspondre à une structure complexe, en combinant requêtes SPARQL et validation par LLM [11]. Ces travaux justifient le choix d’une architecture hybride associant recherche structurée, RAG et LLM.

3.10 Génération augmentée et validation sémantique

Bratanič et Hane soulignent l’intérêt des stratégies de recherche hybride, du *parent document retriever* et des workflows agentiques pour améliorer la précision et l’explicabilité du RAG [1]. Mihindukulasooriya et al. montrent dans leur benchmark *Text2KGBench* que les LLM peinent à garantir la conformité ontologique sans documents de référence [8]. Kampars et al. explorent la conception collaborative d’ontologies assistée par LLM, en associant expertise humaine, mapping sémantique et validation itérative [6]. Dang et al. constatent que 40 à 50% du markup Schema.org généré par GPT est incorrect sans validation formelle, et proposent des agents de curation SHACL [3]. Ces travaux confirment l’intérêt d’une approche RAG ancrée dans des documents de référence et complétée par des mécanismes de validation.

3.11 Outils existants

AI4Semantics

AI4Semantics est un prototype de chatbot de modélisation de données, développé par la Commission européenne dans le cadre de SEMIC. Il est capable d’interroger des

référentiels sémantiques européens et de proposer des classes, attributs et relations. Son code source, organisé en plusieurs dépôts MCP (serveur, client, frontend), a été réutilisé, analysé puis adapté pour intégrer les standards français et d’autres corpus sémantiques.

Autres approches et limites

Plusieurs outils commerciaux ou académiques proposent des assistants à la modélisation de données, à la création d’ontologies ou à la comparaison de schémas. Les travaux récents sur OLaLa, Agent-OM et le complex ontology matching montrent l’intérêt croissant des LLM pour l’alignement sémantique. Cependant, peu d’outils sont adaptés au contexte spécifique des standards publics français, à l’ingestion de documents PDF hétérogènes et à la combinaison de multiples référentiels dans une interface unique. De plus, les solutions propriétaires posent des questions de souveraineté, de coût et d’impact environnemental.

3.12 Positionnement du stage

Ce stage se situe à l’intersection de plusieurs domaines : l’interopérabilité sémantique, l’ingénierie des connaissances, l’intelligence artificielle appliquée aux données publiques et la frugalité numérique. Il reprend les principes du RAG et des agents LLM pour proposer une solution dédiée au contexte du CNIG et des standards de données français. L’originalité du travail réside dans l’intégration de multiples corpus sémantiques (CNIG, schema.data.gouv.fr, LOV, INSPIRE, FIWARE, schema.org, SAREF, SEMIC) dans un même système RAG, le passage d’un chatbot à un outil de modélisation interactif, et la prise en compte des contraintes de souveraineté et de frugalité dans les choix techniques.

Chapitre 4

Approche suivie et solution proposée

4.1 Analyse du besoin

Le besoin initial du stage était d'évaluer la pertinence d'utiliser l'intelligence artificielle pour accompagner les groupes de travail du CNIG dans la standardisation et la mise en cohérence de modélisations de données. Plus concrètement, il s'agissait de :

- fournir aux rédacteurs de standards un accès facilité aux modélisations existantes ;
- éviter la recreation de concepts déjà définis dans d'autres standards ;
- permettre l'alignement sémantique entre référentiels français, européens et internationaux ;
- proposer une interface unique et accessible, à la fois conversationnelle et de modélisation.

Les hypothèses de travail étaient les suivantes : un LLM assisté par une architecture RAG peut comprendre un besoin de modélisation exprimé en langage naturel, identifier les standards pertinents et proposer une structuration cohérente. L'outil doit pouvoir fonctionner avec des documents PDF, intégrer plusieurs corpus hétérogènes et rester déployable dans un environnement souverain.

4.2 Conception de la solution

Rappel sur la modélisation de données

La modélisation de données consiste à définir les **classes**, **attributs** et **relations** d'un domaine de manière structurée et interopérable. Dans le contexte des standards publics, elle s'inscrit dans une démarche collaborative où des groupes de travail produisent un **schéma de données** ou une **ontologie**. Les principaux formalismes rencontrés sont UML (diagrammes de classes), OWL/RDF, Turtle/JSON-LD et XMI pour l'échange entre outils. L'étape la plus coûteuse est l'analyse de l'existant : les modélisateurs doivent consulter de nombreux documents hétérogènes, souvent rédigés dans des vocabulaires différents. HarmonIA vise à lever cette difficulté en fournissant un assistant IA capable d'explorer automatiquement les référentiels et de proposer des alignements sémantiques.

Architecture globale

La solution repose sur une architecture en trois couches :

1. **Couche d'ingestion et d'indexation** : les documents des différents corpus sont téléchargés, découpés en fragments, encodés sous forme d'embeddings denses et sparse, puis stockés dans une base vectorielle.
2. **Couche de recherche et de reranking** : lors d'une requête, le système recherche les fragments les plus proches sémantiquement, les réordonne avec un modèle de reranking, puis sélectionne les meilleurs résultats.
3. **Couche de génération et d'interaction** : un LLM reçoit la requête utilisateur enrichie des documents pertinents. Il peut converser, proposer des modélisations, et manipuler un modèle via des outils MCP.

La figure 4.1 illustre la structure fonctionnelle de HarmonIA. L'utilisateur interagit avec un orchestrateur LLM, qui planifie et exécute des appels à une boîte à outils composée d'un planificateur, d'un moteur de recherche documentaire, d'outils de contrôle/validation et d'outils d'édition du schéma de données.

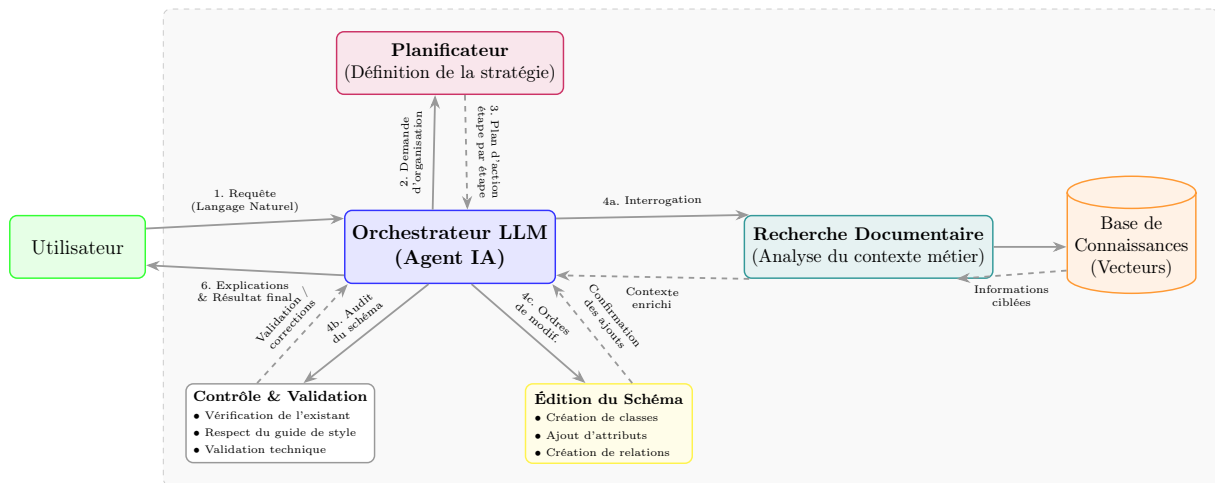


FIGURE 4.1 – Architecture fonctionnelle de HarmonIA : l'orchestrateur LLM planifie et exécute des appels à une boîte à outils (recherche, validation, édition).

Modélisation du système RAG

La recherche sémantique de HarmonIA repose sur une combinaison de techniques éprouvées en recherche d'information et en RAG. Les documents sont d'abord découpés en fragments (*chunks*) avec recouvrement, puis indexés dans Qdrant sous forme d'embeddings denses et sparse grâce au modèle BGE-M3 [2]. Cette dualité permet de capturer à la fois la similarité sémantique générale et la correspondance lexicale exacte, particulièrement utile pour les termes rares, les acronymes ou les identifiants techniques présents dans les standards.

Le score de pertinence initial est obtenu par une combinaison linéaire normalisée des scores dense et sparse :

$$S_{hybride} = 0,7 \times S_{dense} + 0,3 \times S_{sparse} \quad (4.1)$$

Le coefficient 0,7 privilégie la sémantique générale tout en conservant une contribution significative du signal lexical. Cette valeur est cohérente avec les recommandations issues de l'évaluation de BGE-M3 sur le benchmark MIRACL, qui suggèrent un ratio d'environ 77% dense / 23% sparse.

Après la récupération d'un pool de candidats par Qdrant, un modèle cross-encoder BGE-Reranker-v2-M3 ré-évalue chaque couple (requête, fragment) de manière plus fine qu'un simple produit scalaire. Cette architecture *retrieve then rerank* est un standard industriel formalisé notamment par Pradeep et al. [9] et par ColBERT [7]. Dans HarmonIA, le reranker est appliqué en deux étapes : d'abord sur le meilleur fragment de chaque document candidat, puis, si la collection a été indexée avec `summary.enabled=true`, sur le résumé du document (chunk 0) des meilleurs candidats. Le score final est alors la moyenne équipondérée des deux scores, croisant pertinence locale du fragment et cohérence globale du document.

Les scores de plusieurs fragments appartenant au même document sont agrégés selon une combinaison pondérée du meilleur score et de la moyenne des trois meilleurs scores :

$$S_{doc} = 0,9 \times \max(S_{chunks}) + 0,1 \times \text{moyenne}(\text{top_3}(S_{chunks})) \quad (4.2)$$

Cette agrégation reflète l'idée qu'un document est pertinent s'il contient au moins un passage très pertinent, tout en pénalisant légèrement les documents avec un seul pic isolé.

La figure 4.2 résume le pipeline de recherche sémantique de HarmonIA. La partie supérieure correspond à la phase d'**indexation hors ligne** : les documents sont découpés en fragments, encodés sous forme d'embeddings denses et sparse, puis stockés dans Qdrant. La partie inférieure correspond à la phase de **requête en ligne** : la question de l'utilisateur est vectorisée, les fragments les plus proches sont récupérés, réordonnés par un modèle cross-encoder, puis injectés dans le contexte du LLM pour générer une réponse ancrée dans les sources.

Format pivot des modèles de données

Quelle que soit la source d'import (XMI/UML, OWL/RDF Turtle, JSON natif, SQL), le serveur normalise le modèle vers une structure pivot commune contenant des `elements`, des `connectors`, une vue JSON-LD (`ttl`) et une vue XMI (`xmi`). Ce format pivot permet aux outils MCP de travailler de manière uniforme, quelle que soit l'origine du modèle. Le mapping conceptuel entre les différentes représentations est résumé ci-dessous.

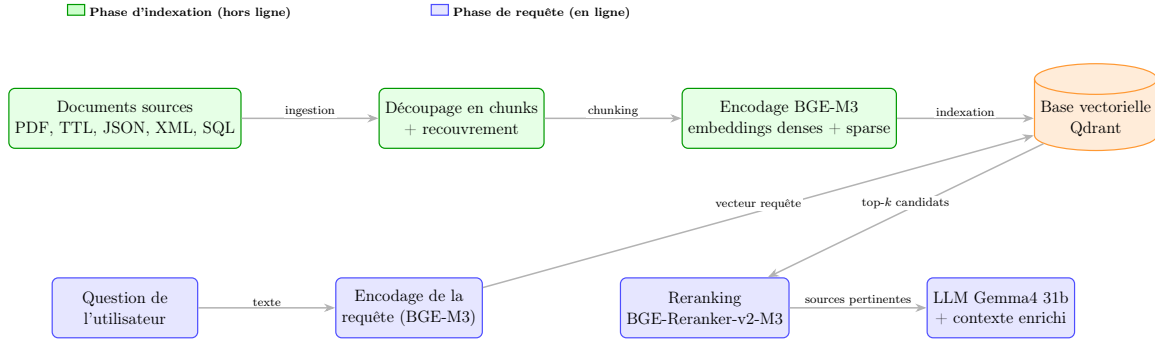


FIGURE 4.2 – Pipeline RAG de HarmonIA. En phase d’indexation (haut), les documents sont découpés, encodés et stockés dans Qdrant. En phase de requête (bas), la question est encodée, envoyée à Qdrant, les candidats sont réordonnés par le reranker, puis injectés dans le LLM.

Concept pivot	RDF / OWL	XMI / UML	SQL
Package	owl:Ontology	uml:Package	Schéma / base
Classe	owl:Class	uml:Class	Table
Attribut	owl:DatatypeProperty	uml:Property	Colonne
Association	owl:ObjectProperty	uml:Association	Clé étrangère
Généralisation	rdfs:subClassOf	Généralisation UML	—
Libellé / définition	rdfs:label, rdfs:comment	Tags EA / notes	Commentaires

TABLE 4.1 – Correspondance entre le format pivot et les représentations RDF, UML et SQL

Les identifiants sont déterministes (UUID5 sur les URIs), ce qui permet de mettre à jour un élément existant sans le dupliquer et de maintenir la cohérence entre les vues XMI et RDF du modèle.

Choix technologiques

Les principaux choix technologiques sont résumés dans le tableau ci-dessous.

4.3 Développement et implémentation

Appropriation et correction du code AI4Semantics

Le point de départ du développement était le code du prototype **AI4Semantics** de la Commission européenne, structuré selon l’architecture MCP (*Model Context Protocol*) en un serveur d’outils sémantiques, un client d’orchestration et un frontend web. La première tâche a consisté à en comprendre l’architecture, les dépendances et le fonctionnement interne. Plusieurs problèmes ont été identifiés et corrigés :

- **Gestion de l’historique** : l’outil conservait l’intégralité des résultats des appels

Composant	Technologie	Justification
LLM	Gemma4 31b via Ol-lama Cloud	Hébergement européen, non conservation des requêtes, bonne stabilité agentique
Embeddings	BGE-M3	Embeddings denses et sparse multilingues dans un seul modèle
Reranking	BGE-Reranker-v2-M3	Réordonnancement cross-encoder performant
Base vectorielle	Qdrant	Open source, compatible Docker et SSPCloud
Backend client	FastAPI + frontend statique	API REST, authentification, déploiement souverain
Backend serveur	FastMCP + Uvicorn	Protocole MCP standard pour exposer les outils
Graphe RDF	<code>rdflib</code>	Parsing/synchronisation Turtle/OWL/JSON-LD
Validation	API SHACL ITB	Validation formelle contre le SEMIC Style Guide
Visualisation	PlantUML natif	Génération locale de diagrammes de classes
Déploiement	SSPCloud Onyxia	Plateforme souveraine des services statistiques publics

TABLE 4.2 – Choix technologiques principaux

aux outils et des documents injectés, ce qui conduisait à un dépassement rapide de la fenêtre de contexte du LLM. Un mécanisme de réduction et de sélection de l'historique a été mis en place.

- **Optimisation des documents remontés** : les fragments remontés par la recherche vectorielle n'étaient pas toujours les plus pertinents. L'ajout du reranking et de la combinaison dense/sparse a permis d'améliorer la qualité des sources.

Intégration des corpus de standards

L'index Qdrant est alimenté par un corpus structuré placé dans le dossier `documents/`. L'arborescence de ce dossier détermine les *tags* associés à chaque point indexé, ce qui permet un filtrage dynamique par source ou par thème dans l'interface utilisateur. Le corpus final intégré dans le cadre du stage est composé de plus de **5 000 fichiers**, dont **16 PDF** (les standards du CNIG) et environ **5 000 fichiers structurés** (JSON, TTL, XML, XMI). Les sources représentées sont les suivantes :

- standards du CNIG, majoritairement en PDF, dans le dossier `schema.data.gouv.fr/CNIG/` ;
- schémas de données publiés sur `schema.data.gouv.fr` ;
- modèles SEMIC de la Commission européenne (Core Vocabularies, etc.) ;
- référentiels complémentaires : Linked Open Vocabularies (LOV), INSPIRE, FI-WARE, schema.org, SAREF.

Le script d'indexation `load.py` supporte plusieurs formats (JSON, XML/XMI, TTL/OWL, SQL, Markdown, PDF) et adapte le découpage (*chunking*) à chaque type. Les documents PDF du CNIG, particulièrement hétérogènes, sont convertis en Markdown via `pymupdf4llm` avant découpage, ce qui préserve mieux la structure sémantique (titres, listes, tableaux) qu'une extraction textuelle brute. Les descriptions natives (`rdfs:comment`,

`dterms:description`, etc.) sont extraites lorsqu’elles sont disponibles ; sinon, un résumé automatique peut être généré par LLM.

Passage d’un chatbot à un outil de modélisation

L’interface web de HarmonIA est une application monopage (SPA) organisée autour de trois outils principaux, accessibles depuis la barre de navigation.

Chercher. Cet onglet propose un moteur de recherche sémantique directe dans le corpus Qdrant, avec filtrage par tags/sources et ajout des documents pertinents au contexte de l’assistant (figure 4.3).



FIGURE 4.3 – Interface de chercher des modèles dans HarmonIA.

Éditer. Cet environnement permet d’importer un modèle (XMI/UML, OWL/TTL, JSON, SQL, texte), de le visualiser sous forme de diagramme PlantUML, de l’éditer et de l’exporter en XMI ou TTL. La figure 4.4 montre l’analyse d’un modèle importé.

Analyser. Cet onglet est une interface conversationnelle classique : l’utilisateur dialogue en langage naturel et l’agent LLM appelle les outils MCP en arrière-plan pour répondre. Il est possible d’importer jusqu’à trois modèles simultanément afin de les interroger, de les fusionner ou de les analyser ; cette limite vise à préserver la fenêtre de contexte du LLM, qui serait saturée par des modèles trop volumineux. Un exemple de conversation est présenté dans le chapitre 5 (figures 5.1 et 5.2).

Cette évolution a permis de transformer l’outil initial, limité à un chatbot de questions-réponses, en une plateforme de modélisation conversationnelle. Les outils MCP exposés au modèle incluent :

- `retrieve_documents` : recherche sémantique dans le corpus indexé ;

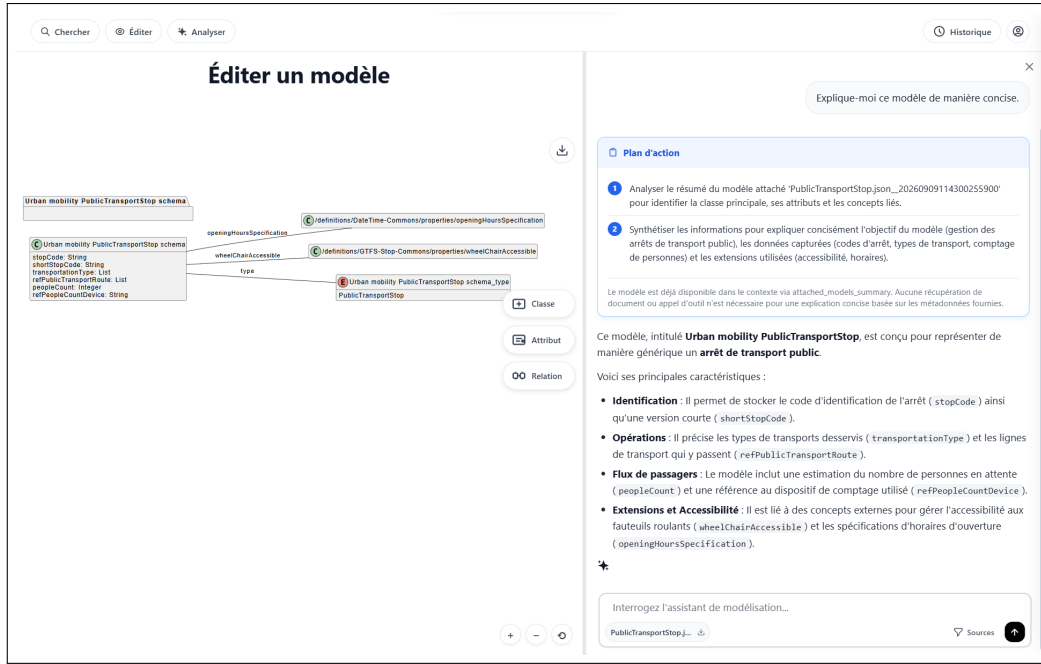


FIGURE 4.4 – Onglet d’édition : analyse d’un modèle importé.

- `metadata_checker` : vérification de la complétude et de la cohérence des métadonnées selon les règles GC-R3, GC-R4, GC-R5 et GC-R7 du SEMIC Style Guide ;
- `reuse_check` : évaluation de la réutilisation des concepts existants pour chaque classe du modèle ;
- `validator_check` : validation SHACL du modèle via l’API ITB de la Commission européenne ;
- `style_guide_check` : génération d’un rapport Markdown d’évaluation globale ;
- `add_class`, `add_attribute`, `add_connector` : mutations du modèle sous contrôle de l’utilisateur.

L’agent planificateur `plan_workflow_with_tools` est appelé au début de chaque tour de chat pour décomposer la requête en un plan structuré, sauf exceptions définies dans le prompt système. Le plan précise les outils à appeler, l’ordre des appels et les observations attendues. L’agent exécuter réalise ensuite les appels d’outils, met à jour le modèle et formule une réponse synthétique.

Difficultés rencontrées et choix effectués

La principale difficulté technique a été de reprendre un code développé par une autre équipe, avec des pratiques et des choix techniques qu’il fallait comprendre avant de pouvoir l’étendre. Le contact avec l’équipe SEMIC a permis d’échanger sur les retours utilisateurs, mais les aspects techniques profonds étaient limités car le développement avait été externalisé.

Une autre difficulté a résidé dans le traitement des PDF du CNIG : formats variés,

structures irrégulières, tableaux et schémas. Le découpage automatique a dû être affiné pour conserver la cohérence sémantique des fragments. La conversion PDF vers Markdown via `pymupdf4llm` a permis d'améliorer la qualité des fragments extraits par rapport à une extraction textuelle brute.

Enfin, la gestion de la fenêtre de contexte a constitué un point d'attention constant. En réduisant l'historique, en sélectionnant les documents les plus pertinents et en optimisant les prompts, il a été possible de maintenir des conversations complètes sans dépasser les limites de tokens.

Sécurité et déploiement

Le déploiement sur SSPCloud Onyxia a permis de rendre l'outil accessible via une URL publique tout en conservant un hébergement sur une plateforme souveraine. Plusieurs éléments de sécurité ont été mis en place :

- **Authentification locale** : les utilisateurs s'inscrivent et se connectent via `auth_service.py`; les mots de passe sont hachés avec `bcrypt` avant stockage en base SQLite.
- **Clés API externes** : l'endpoint `/api/external/v1` permet un accès programmatique via des clés API Bearer. Seul un hash de chaque clé est conservé en base; la clé en clair n'est affichée qu'une seule fois à la création.
- **Souveraineté des données** : le LLM utilisé pour le POC est hébergé en Europe via Ollama Cloud, sans conservation des requêtes. Le serveur PlantUML est installé localement pour éviter l'envoi de diagrammes à un service distant.
- **Isolation des utilisateurs** : les modèles, historiques de recherche et conversations sont associés à un `user_id`. Une revue de sécurité complète reste nécessaire avant une exposition publique large.

Différences entre le projet SEMIC initial et la solution HarmonIA

Le tableau ci-dessous synthétise les principales évolutions apportées par rapport au projet AI4Semantics initial.

Ces évolutions traduisent un passage d'un prototype conversationnel à une plateforme plus complète, souveraine et adaptée au contexte français des standards de données publiques.

Aspect	AI4Semantics initial	HarmonIA (version développée)
Interface utilisateur	Streamlit simple	FastAPI + SPA vanilla JS, Tailwind CSS
Backend LLM	OpenAI / AzureOpenAI	API OpenAI-compatible, modèles ouverts via Ollama Cloud
Recherche sémantique	Embedding dense simple avec Qdrant	Hybrid search dense + sparse (BGE-M3) + reranking (BGE-Reranker-v2-m3)
Gestion des documents	Indexation textuelle limitée	Support PDF, Markdown, XML, TTL, JSON, SQL avec découpage optimisé
Outils de modélisation	Chatbot conversationnel	Outils MCP + interface de modélisation (classes, attributs, relations)
Authentification	Aucune	Authentification par session, clés API SHA-256, historique SQLite
Visualisation	PlantUML via service externe	PlantUML natif, export SVG/PNG/XMI/TTL
Déploiement	Local via Docker	SSPCloud Onyxia, API externe exposée
Corpus intégrés	Core Vocabularies SEMIC, DCAT-AP, OSLO	CNIG, schema.data.gouv.fr, SEMIC, LOV, INSPIRE, FIWARE, schema.org, SAREF

TABLE 4.3 – Comparaison entre le projet AI4Semantics initial et la solution HarmonIA

Chapitre 5

Validation

5.1 Protocole et résultats de validation

La validation du stage s’est appuyée sur des tests fonctionnels internes, des conversations complètes de modélisation et des retours qualitatifs auprès des tuteurs, de l’équipe Ecolab, du CNIG et de partenaires institutionnels, notamment dans le cadre du projet **Building Data X**. Les critères retenus étaient la pertinence des documents remontés, la capacité à citer ses sources, la cohérence des modélisations proposées, la stabilité des conversations et la facilité de prise en main.

Amélioration de la pertinence du RAG

L’ajout de la recherche hybride dense/sparse et du reranking a sensiblement amélioré la qualité des sources remontées. Sur les questions thématiques posées aux partenaires, un standard pertinent figurait systématiquement dans le top 5 des documents retournés.

Observations quantitatives. En production sur SSPCloud Onyxia :

- **Recherche avec reranking API Albert** : moins de **3 secondes** par requête.
- **Recherche avec reranking local (CPU)** : entre **10 et 15 secondes** par requête, avec une allocation de **15 000 à 30 000 millicpus** sur SSPCloud.

Albert est l’infrastructure publique d’intelligence artificielle générative de l’État, développée par Etalab au sein de la Direction interministérielle du numérique (DINUM). Hébergée dans un environnement sécurisé et souverain (SecNumCloud), elle est accessible aux administrations via une API. Elle est utilisée ici pour le reranking des documents, ce qui permet d’obtenir une latence faible tout en conservant une qualité de réordonnement élevée. Ces résultats illustrent le compromis classique entre latence et souveraineté : l’API Albert offre rapidité et qualité, tandis que le reranking local garantit que les données ne quittent pas l’environnement hébergé.

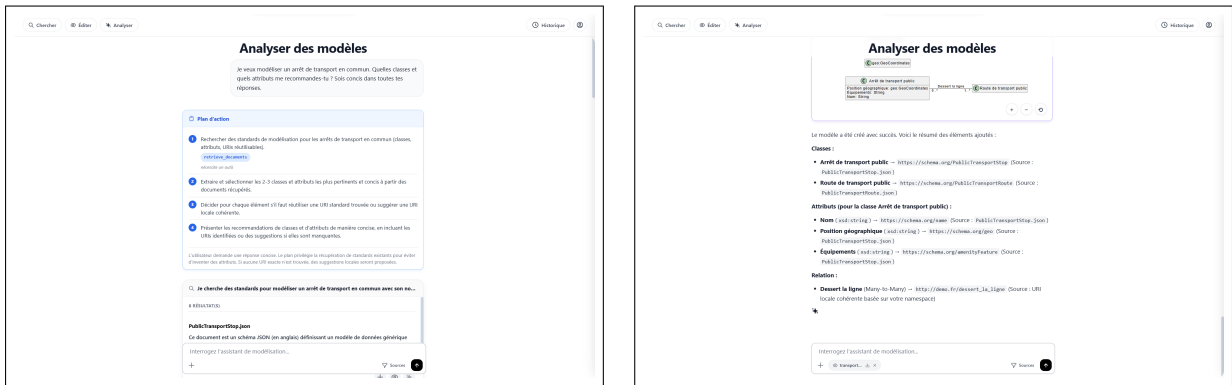
Conversations et modélisation interactive

Après correction de la gestion de l’historique, l’outil a maintenu des conversations complètes sans dépassement de la limite de contexte. L’interface finale propose une recherche sémantique directe, une visualisation du modèle en cours, et des outils d’ajout de classes,

d'attributs et de relations. Cette transformation d'un simple chatbot en plateforme de modélisation a été particulièrement bien accueillie.

Exemple de conversation de modélisation

La figure 5.1 illustre une conversation complète réalisée avec HarmonIA sur le thème des arrêts de transport public. L'utilisateur demande d'abord quelles classes et attributs sont recommandés pour modéliser un arrêt de transport en commun, en demandant des réponses concises. L'assistant élabore un plan d'action (figure 5.1a), recherche des standards pertinents, propose des recommandations, puis crée un modèle cohérent avec les référentiels existants. La figure 5.1b présente le résumé final du modèle construit, avec les classes, attributs, relation et URIs standard réutilisées.



(a) Début de la conversation : plan d'action proposé par l'agent.

(b) Fin de la conversation : résumé du modèle de données construit.

FIGURE 5.1 – Exemple de conversation de modélisation dans HarmonIA.

La figure 5.2 présente les étapes intermédiaires de la même conversation. Après la première réponse synthétique, l'assistant élabore un plan d'action pour la création du modèle, vérifie la réutilisation d'URIs standards, obtient la confirmation de l'utilisateur et construit progressivement le modèle avec un diagramme visuel.

Retours des partenaires institutionnels

Les retours qualitatifs ont confirmé l'intérêt de l'outil :

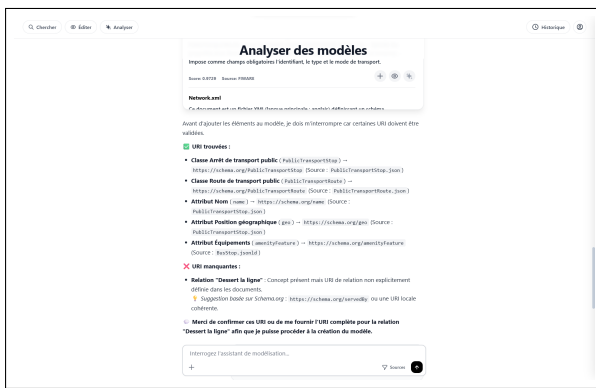
- la DINUM et data.gouv.fr se sont montrées intéressées, sans cas d'usage immédiat sur les jeux de données bruts ;
- OpenData France a manifesté un intérêt marqué pour une collaboration future ;
- les acteurs de **Building Data X**, notamment Orange Business, ont été les plus enthousiastes. Leur test a montré une complémentarité entre HarmonIA et leur générateur de taxonomies (DOG), avec un raisonnement plus clair et une meilleure optimisation des tokens.



(a) Réponse initiale : classe, attributs et relation suggérés à partir des documents FIWARE.



(b) Plan d'action de création : recherche d'URIs standards et confirmation utilisateur.



(c) Proposition de réutilisation d'URIs standards et identification des URI manquantes.



(d) Création du modèle et diagramme visuel généré.

FIGURE 5.2 – Étapes intermédiaires de la conversation de modélisation.

Livrables produits

Les livrables remis comprennent le code source sur GitHub, une documentation technique, une API externe, une interface web déployée sur SSPCloud Onyxia, ainsi qu'un diaporama et un formulaire de retour utilisateur.

5.2 Analyse et discussion

Les résultats obtenus valident la faisabilité d'un outil d'IA pour accompagner la standardisation des données au CNIG. L'architecture RAG, couplée à un modèle de langage hébergé de manière souveraine, permet de réduire les hallucinations et d'ancrer les propositions dans des référentiels existants. La combinaison d'embeddings denses et sparse, suivie d'un reranking, s'est avérée pertinente pour un corpus hétérogène comprenant des PDF techniques.

L'évolution vers un outil de modélisation interactive, via le protocole MCP, constitue

une avancée notable par rapport à un simple chatbot. Elle ouvre la voie à des usages concrets de co-conception de standards, en complément des outils existants comme Enterprise Architect.

Limites identifiées :

- le traitement des PDF reste perfectible : tableaux, schémas et structures complexes ne sont pas toujours bien exploités ;
- l'outil ne prend pas encore en charge le matching direct avec des données réelles ou des tables existantes ;
- les évaluations restent principalement qualitatives ; une évaluation quantitative sur un benchmark de modélisation serait souhaitable ;
- l'import de modèles volumineux sature la fenêtre de contexte du LLM : les tokens d'entrée sont entièrement consommés par le modèle importé, rendant impossible tout échange avec l'assistant.

Malgré ces limites, la preuve de concept démontre un potentiel réel pour l'intégration de l'IA dans le processus de standardisation, dans un cadre souverain et frugal.

Chapitre 6

Conclusion et perspectives

6.1 Bilan du stage

Ce stage de fin d'études m'a permis de contribuer à un projet à l'intersection de l'intelligence artificielle, de l'interopérabilité des données et de la transition écologique. Au cours de ces mois chez Ecolab, j'ai mené à bien plusieurs réalisations :

- appropriation, analyse et adaptation du code de l'outil AI4Semantics de la Commission européenne ;
- conception et mise en œuvre d'une architecture RAG multi-corpus intégrant les standards du CNIG, les schémas de `schema.data.gouv.fr` et les référentiels SEMIC, LOV, INSPIRE, FIWARE, `schema.org` et SAREF ;
- amélioration de la pertinence des documents remontés grâce à une combinaison d'embeddings denses/sparse et à un modèle de reranking ;
- transformation d'un chatbot en un outil de modélisation de données assisté par l'IA, via le protocole MCP ;
- déploiement de l'outil sur SSPCloud Onyxia, avec mise à disposition d'une API et d'une interface web ;
- conduite d'échanges avec des partenaires institutionnels et collecte de retours d'usage.

Sur le plan technique, ce stage m'a permis de consolider mes compétences en ingénierie des données, en systèmes RAG, en bases vectorielles, en modèles de langage et en déploiement cloud. J'ai également appris à travailler dans un environnement administratif, à interagir avec des partenaires institutionnels variés et à rendre compte régulièrement de l'avancée du projet.

Sur le plan personnel, cette expérience a confirmé mon intérêt pour les usages de l'IA au service de l'intérêt général, dans un cadre souverain et responsable. L'accueil chaleureux de l'équipe Ecolab et la qualité de l'encadrement ont grandement contribué à la réussite de ce stage.

6.2 Perspectives

Le projet ouvre plusieurs pistes d'évolution prometteuses :

- **Matching avec des données réelles** : intégrer des jeux de données concrets

pour permettre l’alignement automatique de tables ou de fichiers avec les standards existants.

- **Amélioration du traitement des PDF** : exploiter des méthodes d’extraction plus fines, capables de prendre en compte les tableaux, les schémas et les structures imbriquées des documents du CNIG.
- **Évaluation quantitative** : construire un benchmark de modélisation pour mesurer objectivement la qualité des propositions de l’outil.
- **Industrialisation** : poursuivre le développement pour aboutir à un outil opérationnel intégré aux processus de travail des groupes thématiques du CNIG, avec une interface adaptée aux besoins des rédacteurs de standards.
- **Collaborations institutionnelles** : approfondir les échanges avec OpenData France, les acteurs du projet Building Data X (Caisse des Dépôts, Orange Business, Dawex, Docaposte, Sopra Steria, buildingSmart France, FPI), la DINUM, l’AFNOR et Inria pour identifier des cas d’usage concrets et pérennes.
- **Poursuite du partenariat avec Orange Business** : explorer l’intégration de l’outil avec le générateur de taxonomies DOG et la plateforme Dawex dans le cadre de Building Data X.

En conclusion, ce stage a démontré la faisabilité et l’intérêt d’un assistant IA dédié à la standardisation des données géolocalisées. Au-delà du produit développé, il a permis de poser les bases d’une démarche durable, souveraine et frugale pour l’utilisation de l’intelligence artificielle dans l’administration publique.

Bibliographie

- [1] Tomaž Bratanič and Oskar Hane. *Essential GraphRAG : Knowledge Graph–Enhanced RAG*. Manning Publications, 2025. URL : <https://neo4j.com/essential-graph-rag/>.
- [2] Jian Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-embedding : Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv :2402.03216*, 2024. URL : <https://arxiv.org/abs/2402.03216>.
- [3] Minh-Hoang Dang, Thi Hoang Thi Pham, Pascal Molli, Alban Gaignard, and Hala Skaf-Molli. Llm4schema.org : Generating schema.org markups with large language models. *Semantic Web*, 16(6) :1–30, 2025. URL : <https://hal.science/hal-05138664v2/file/dang-et-al-2025-llm4schema-org-generating-schema-org-markups-with-large-language-models.pdf>, doi:10.1177/22104968251382172.
- [4] Sven Hertling and Heiko Paulheim. Olala : Ontology matching with large language models. In *Proceedings of the 11th International Conference on Knowledge Capture (K-CAP '23)*. ACM, 2023. URL : <https://arxiv.org/abs/2311.03837>, doi:10.1145/3587259.3627571.
- [5] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12) :248, 2023. doi:10.1145/3571730.
- [6] Janis Kampars, Guntis Mosans, Tushar Jogi, Franz Roters, and Napat Vajragupta. Llm-supported collaborative ontology design for data and knowledge management platforms. *Frontiers in Big Data*, 8 :1676477, 2025. URL : <https://research.rtu.lv/en/publications/llm-supported-collaborative-ontology-design-for-data-and-knowledg/>, doi:10.3389/fdata.2025.1676477.
- [7] Omar Khattab and Matei Zaharia. Colbert : Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2020. URL : <https://arxiv.org/abs/2004.12832>, doi:10.1145/3397271.3401075.
- [8] Nandana Mihindukulasooriya, Sanju Tiwari, Carlos F. Enguix, and Kusum Lata. Text2kgbench : A benchmark for ontology-driven knowledge graph generation from text. In *The Semantic Web – ISWC 2023*, volume 14266 of *Lecture Notes in Computer*

- Science*, pages 247–265. Springer, 2023. URL : <https://arxiv.org/abs/2308.02357>, doi:10.1007/978-3-031-47243-5_14.
- [9] Ronak Pradeep, Rodrigo Nogueira, and Jimmy Lin. The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2021. URL : <https://arxiv.org/abs/2101.05667>, doi:10.1145/3404835.3462995.
- [10] Zhangcheng Qiang, Weiqing Wang, and Kerry Taylor. Agent-om : Leveraging llm agents for ontology matching. *Proceedings of the VLDB Endowment*, 18(3) :516–529, 2024. URL : <https://arxiv.org/abs/2312.00326>, doi:10.14778/3712221.3712222.
- [11] Ondřej Zamazal. Towards pattern-based complex ontology matching using sparql and llm. In *SEMANTiCS 2024 EU : 20th International Conference on Semantic Systems*. CEUR-WS, 2024. URL : <https://ceur-ws.org/Vol-3759/paper8.pdf>.

Annexes

.1 Annexe A : Codes source / scripts complémentaires

Le code source de la preuve de concept HarmonIA est organisé en deux dépôts GitHub complémentaires :

- **Client web + interface conversationnelle** : <https://github.com/theojoly2/HarmonIA-MCP-client-fr> (FastAPI, frontend statique, authentification, API externe).
- **Serveur MCP + outils sémantiques** : <https://github.com/theojoly2/HarmonIA-MCP-server-fr> (FastMCP, Qdrant, reranking, outils de modélisation, validation SHACL).

Ces dépôts contiennent l'ensemble du code nécessaire pour reproduire le POC, ainsi que les fichiers de configuration Docker et les instructions de déploiement sur SSPCloud Onyxia.

.2 Annexe B : Liste des figures techniques

Les principaux diagrammes et captures d'écran du rapport sont listés ci-dessous.

- Figure 4.1 : architecture fonctionnelle de HarmonIA.
- Figure 4.2 : pipeline de recherche sémantique RAG.
- Figure 4.3 : interface de recherche sémantique.
- Figure 4.4 : onglet de modélisation et analyse d'un modèle importé.
- Figures 5.1 et 5.2 : exemple de conversation de modélisation interactive.

.3 Annexe C : Résultats détaillés

Corpus indexé

Le tableau ci-dessous résume les volumes de fichiers intégrés dans l'index Qdrant final.

Temps de réponse observés

- Recherche avec reranking via l'API Albert : < 3 s.
- Recherche avec reranking local CPU : 10–15 s (15 000–30 000 millipus sur SSPCloud Onyxia).

Source	Nombre de fichiers
FIWARE	1 054
INSPIRE	98
LOV	895
SAREF	16
SEMIC	7
schema.data.gouv.fr	84
schema.org	2 924
CNIG (PDF)	16
Total	5 079

TABLE 1 – Volumes du corpus intégré dans Qdrant (dont 16 PDF du CNIG et 25 fichiers dans le dossier CNIG).

Qualité des résultats

Pour les questions thématiques posées aux partenaires du projet Building Data X, un standard directement pertinent figurait dans les 5 premiers documents remontés. Le reranking a permis de remonter en première position les documents les plus cohérents avec la requête.